DEEP REINFORCEMENT LEARNING EXPLAINED

A Markov Decision Process (MDP) is difined by: $<$S,A,R,γ,p$>$

$\mathcal{S}^+$	set of all states (including terminal states)
$\mathcal{A}$	set of all actions
$\mathcal{R}$	set of all rewards
γ	discount rate (where $0 \leq \gamma \leq 1$)
$p(s', r \mid s, a)$	the one-step dynamics of the environment (transition function), probability of next state s' and reward r, given current state s and current action a $\doteq Pr(S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a)$

Other related definitions

S_t	state at time t
A_t	action at time t
R_t	reward at time t
G_t	discounted return at time t $= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$
$\mathcal{S}$	set of all non-terminal states
$\mathcal{A}(s)$	set of all actions available in state s
Trajectory:	(S_0,A_0, S_1,A_1,R_1, ... , S_{t-1},A_{t-1},R_{t-1},S_t,A_t,R_t)

Policies and Value Function

$\pi(s)$ — deterministic policy
  $\pi(s) \in \mathcal{A}(s)$ for all $s \in \mathcal{S}$

$\pi(a|s)$ — stochastic policy
  $\pi(a|s) = \mathbb{P}(A_t = a | S_t = s)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$

v_π — state-value function for policy π
  $v_\pi(s) \doteq \mathbb{E}[G_t | S_t = s]$ for all $s \in \mathcal{S}$

q_π — action-value function for policy π
  $q_\pi(s,a) \doteq \mathbb{E}[G_t | S_t = s, A_t = a]$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$

v_* — optimal state-value function
  $v_*(s) \doteq \max_\pi v_\pi(s)$ for all $s \in \mathcal{S}$

q_* — optimal action-value function
  $q_*(s,a) \doteq \max_\pi q_\pi(s,a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$

$\pi_*(s)$ — Optimal policy
  $\pi_*(s) \doteq \arg\max_{a \in \mathcal{A}(s)} q_*(s,a)$

Bellman Equations

$$v_\pi(s) = \sum_{a \in \mathcal{A}(s)} \pi(a|s) \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r|s, a)(r + \gamma v_\pi(s'))$$

$$q_\pi(s,a) = \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r|s, a)(r + \gamma \sum_{a' \in \mathcal{A}(s')} \pi(a'|s') q_\pi(s', a'))$$

$$v_*(s) = \max_{a \in \mathcal{A}(s)} \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r|s, a)(r + \gamma v_*(s'))$$

$$q_*(s,a) = \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r|s, a)(r + \gamma \max_{a' \in \mathcal{A}(s')} q_*(s', a'))$$

First-Visit MC Prediction Algorithm

Algorithm 1: First-Visit MC Prediction

Input: policy π, *num_episodes*

Initialize $N(s,a) = 0$ for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$

Initialize $returns\_sum(s,a) = 0$ for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$

for $i \leftarrow 1$ ***to*** *num_episodes* **do**

  Generate an episode $S_0, A_0, R_1, \ldots, S_T$ using π

  for $t \leftarrow 0$ ***to*** $T-1$ **do**

    if (S_t, A_t) *is a first visit (with return* G_t*)* **then**

      $N(S_t, A_t) \leftarrow N(S_t, A_t) + 1$

      $returns\_sum(S_t, A_t) \leftarrow returns\_sum(S_t, A_t) + G_t$

  end

end

$Q(s,a) \leftarrow returns\_sum(s,a)/N(s,a)$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

return Q

$$\pi_0 \xrightarrow{E} q_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} q_{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} ... \xrightarrow{I} \pi_* \xrightarrow{E} q_{\pi_*}$$

$$S_0, A_0, R_1, \ldots, S_T$$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(G_t - Q(S_t, A_t))$$

1-$\epsilon + \epsilon/|A(S)|$

$\epsilon/|A(S)|$

Constant-α MC Control with ϵ decay

Algorithm 2: Constant-α MC Control with ϵ decay

Input: *num_episodes*, α, ϵ-decay, γ
Initialize $Q(s, a) = 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$
for $i \leftarrow 1$ ***to*** *num_episodes* **do**
  $\epsilon \leftarrow$ setting new epsilon with ϵ-decay
  $\pi \leftarrow \epsilon$-greedy(Q)
  Generate an episode $S_0, A_0, R_1, \ldots, S_T$ using π
  for $t \leftarrow 0$ ***to*** $T - 1$ **do**
    $G_t \leftarrow$ compute discounted return using γ
    $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(G_t - Q(S_t, A_t))$
  end
end
return π

MC and TD Update equations summary

Monte Carlo $\quad Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(\ \boldsymbol{G_t}$ - Q(S$_t, A_t))$

Sarsa $\quad Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(\ \boldsymbol{R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})}\ - Q(S_t, A_t))$

Sarsamax $\quad Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(\ \boldsymbol{R_{t+1} + \gamma \max_{a \in \mathcal{A}(s)} Q(S_{t+1}, a)}\ - Q(S_t, A_t))$

Expected Sarsa $\quad Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(\ \boldsymbol{R_{t+1} + \gamma \sum_{a \in \mathcal{A}(s)} \pi(a|S_{t+1}) Q(S_{t+1}, a)}\ - Q(S_t, A_t))$

Stub for the post

S_0, A_0, R_1, S_1, A_1

Sarsa (t=0) $Q(S_0, A_0) \leftarrow Q(S_0, A_0) + \alpha(\boldsymbol{R_1 + \gamma Q(S_1, A_1)} - Q(S_0, A_0))$

(*) For further references go to the next url: `https://torres.ai/deep-reinforcement-learning-explained-series`